

# SHAKE, Rattle, and Roll: Efficient Constraint Algorithms for Linked Rigid Bodies

TIMOTHY R. FORESTER, WILLIAM SMITH

*Department for Computation and Information, CCLRC Daresbury Laboratory, Daresbury, Warrington, Cheshire WA4 4AD, United Kingdom*

*Received 28 October 1996; accepted 25 August 1997*

**ABSTRACT:** We present an iterative constraint algorithm, QSHAKE, for use with semirigid molecules in molecular dynamics simulations. The algorithm is based on "SHAKE-ing" bond constraints between rigid bodies, whose equations of motion are solved in the quaternion framework. The algorithm is derived and its performance compared with SHAKE for liquid octane. QSHAKE is significantly more efficient whenever SHAKE requires triangles (or tetrahedra) of constraints to maintain molecular shape. Efficiencies are gained because QSHAKE reduces the number of holonomic constraints that must be solved iteratively and requires fewer iterations to obtain convergence. The gains in efficiency are most noticeable when a high degree of precision is imposed on the constraint criteria. QSHAKE is also stable at larger time steps than SHAKE, thus allowing for even more efficient exploration of phase space for semirigid molecules. © 1998 John Wiley & Sons, Inc. *J Comput Chem* **19**: 102–111, 1998

**Keywords:** QSHAKE; bond constraints; semirigid molecules; molecular dynamics simulations

## Introduction

Over the past decades many in the molecular simulation field have wrestled with the issue of how to integrate the equations of motion for rigid and semirigid molecules. Rigid molecules (or

rigid bodies) for our purposes are collections of point atoms whose local geometry is time invariant; a "semirigid" molecule is taken to be a series of rigid bodies linked by bonds of fixed length such that there are some "internal" (i.e., bond angle and torsional) degrees of freedom in the molecule.

Early attempts at simulating rigid molecules involved solving the Euler equations,<sup>1,2</sup> but attention later switched to quaternion algorithms,<sup>3</sup> which avoided the singularity in the Euler method. The introduction of an implicit quaternion al-

Correspondence to: W. Smith; e-mail: w.smith@daresbury.ac.uk

gorithm<sup>4</sup> provided a scheme that was stable at relatively large time steps and thus allowed for more efficient exploration of phase space. An alternate approach, and by far the most popular for semirigid molecules, is constraint dynamics.<sup>5</sup> In this approach rigid molecular subunits are defined by applying holonomic constraints to specified interatomic separations. In practice this means solving the unconstrained equations of motion and retrospectively determining the constraint forces that need to be applied at the beginning of the time step to ensure that the bond lengths are maintained at constant length. This approach has become standard in the molecular simulation of complex molecules.<sup>6</sup> For small molecules the linearized constraint equations can be solved algebraically, but for large molecules this becomes prohibitively expensive as the inversion of a  $c \times c$  matrix (where  $c$  is the number of constraints) is required at each time step. Alternatively, the constraint forces can be found by iteratively approximating each constraint force in turn until self-consistency is obtained. Iterations continue until each bond is within some preset tolerance of the specified bond lengths. This approach is at the heart of the SHAKE<sup>5</sup> and RATTLE<sup>7</sup> algorithms. (SHAKE operates on atomic positions while RATTLE operates on both atomic positions *and* velocities.) Bond constraints are most commonly used to remove high frequency modes (such as C—H or C—C stretches) from a system to allow the use of a larger time step and to prevent energy from being locked into modes that are not well coupled to the rest of the system.

However, the network of constraint bonds required by SHAKE (or RATTLE) for rigid and semirigid molecules is often problematic. The SHAKE iterations converge slowly whenever rigid triangles of constraints are present. The consequence is that reduced time steps may be required and that the SHAKE algorithm can become a significant consumer of CPU time. Triangles of constraint bonds are needed to define rigid planar units (as occurs in phenyl rings and peptide groups) or 3-dimensional (3-D) units (e.g., rigid tetrahedral conformations). (The converse is also true: any triangle of constraint bonds defines a rigid unit.) The convergence of SHAKE is even more problematic when these units are linked by other constraint bonds.

In this article we address the problem of linked rigid units, using an extension of the SHAKE algorithm we call QSHAKE that incorporates the quaternion rotational description for rigid bodies

to provide an efficient constraint algorithm for molecular models of this class. In presenting our method, we note that a general algorithm already exists for such systems, that of Kneller and Hinsen,<sup>8</sup> which is based on the Lagrangian formulation. This method has the interesting capability of modeling rigid units linked by a single shared atom (i.e., a “pivot”) or linked via a shared constraint bond (i.e., a “hinge”), which is not generally within the capability of QSHAKE. However, nevertheless our method has applicability to a wide range of molecular structures while retaining the conceptual simplicity of SHAKE, which is almost universally employed in current macromolecular simulations, and for which it offers significant efficiency improvements. For highly connected networks QSHAKE can be an order of magnitude faster than SHAKE as well as being stable at longer time steps.

We begin our description with a brief review of the quaternion leapfrog algorithms and SHAKE. QSHAKE is then derived and finally tested against SHAKE for three different models of liquid octane.

## Quaternion Framework

A rigid body has associated with it a rotational inertia tensor  $\underline{\mathbf{I}}$ , whose components are given by  $I_{\alpha\beta} = \sum_{\text{sites}} m s_{\alpha} s_{\beta}$ , where the  $s$  are the distance to the center of mass and the  $m$  are the site masses. The local body frame is taken to be that in which the rotational inertia tensor  $\underline{\mathbf{I}}$  is diagonal. The orientation of a local body frame with respect to the space fixed frame is described via a 4-D unit vector, the quaternion  $q = [q_0, q_1, q_2, q_3]^{\dagger}$  (where the dagger denotes the transpose). The rotational matrix to transform from the space fixed frame to the local body frame is<sup>9</sup>

$$\underline{\mathbf{R}} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 & 2(q_1q_2 - q_0q_3) \end{pmatrix}. \quad (1)$$

$\underline{\mathbf{R}}$  is unitary ( $\underline{\mathbf{R}}^{-1} = \underline{\mathbf{R}}^{\dagger}$ ), so if  $\hat{s}_{\alpha}$  (where the circumflex denotes a vector or tensor in the local

body frame) is the position of a site with respect to its center of mass (COM), its position in the space fixed frame (with regard to its COM) is given by the transformation

$$\underline{s}_\alpha = \underline{\mathbf{R}}^\dagger \cdot \underline{\hat{s}}_\alpha. \quad (2)$$

The net translational force acting upon a rigid unit is

$$\underline{F} = \sum_\alpha \underline{f}_\alpha, \quad (3)$$

where  $\underline{f}_\alpha$  is the force on a rigid unit site and the sum includes all sites  $\alpha$  in the body. The translational motion can be integrated by the standard leapfrog algorithm.

$$\underline{V}\left(t + \frac{\Delta t}{2}\right) = \underline{V}\left(t - \frac{\Delta t}{2}\right) + \Delta t M^{-1} \underline{F}(t), \quad (4)$$

$$\underline{R}(t + \Delta t) = \underline{R}(t) + \Delta t \underline{V}\left(t + \frac{\Delta t}{2}\right), \quad (5)$$

where  $M$  is the mass of the rigid unit,  $\underline{R}$  is the body's COM position, and  $\underline{V}$  is the rigid body's COM velocity.

The torque acting upon the body in the space fixed frame is

$$\underline{\tau} = \sum_\alpha \underline{s}_\alpha \times \underline{f}_\alpha. \quad (6)$$

Transformed to the local body frame this is

$$\underline{\hat{\tau}} = \underline{\mathbf{R}} \cdot \underline{\tau}. \quad (7)$$

The angular velocity transformed to the local body frame,  $\underline{\hat{\omega}}$ , can then be integrated using the leapfrog algorithm and the diagonal rotational inertia tensor.

$$\underline{\hat{\omega}}\left(t + \frac{\Delta t}{2}\right) = \underline{\hat{\omega}}\left(t - \frac{\Delta t}{2}\right) + \Delta t \underline{\hat{\mathbf{I}}}^{-1} \cdot \underline{\hat{\tau}}(t). \quad (8)$$

The new quaternions cannot be found so simply. The most efficient and stable scheme for obtaining them is the implicit algorithm due to Fincham.<sup>4</sup> In this algorithm the new quaternions are found by solving the implicit equation

$$\underline{q}(t + \Delta t) = \underline{q}(t) + \frac{\Delta t}{2} \left\{ \underline{\mathbf{Q}}[\underline{q}(t)] \cdot \underline{\hat{\omega}}(t) + \underline{\mathbf{Q}}[\underline{q}(t + \Delta t)] \cdot \underline{\hat{\omega}}(t + \Delta t) \right\}, \quad (9)$$

where  $\underline{\hat{\omega}} = [0, \underline{\hat{\omega}}]^\dagger$ , for which the full step angular velocities are obtained, by analogy with (8), as

$$\underline{\hat{\omega}}(t) = \underline{\hat{\omega}}\left(t - \frac{\Delta t}{2}\right) + \frac{\Delta t}{2} \underline{\hat{\mathbf{I}}}^{-1} \cdot \underline{\hat{\tau}}(t), \quad (10)$$

$$\underline{\hat{\omega}}(t + \Delta t) = \underline{\hat{\omega}}(t) + \Delta t \underline{\hat{\mathbf{I}}}^{-1} \cdot \underline{\hat{\tau}}(t), \quad (11)$$

and  $\underline{\mathbf{Q}}[\underline{q}]$  is

$$\underline{\mathbf{Q}} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}. \quad (12)$$

The above equation (9) is solved iteratively with

$$\underline{q}(t + \Delta t) = \underline{q}(t) + \Delta t \underline{\mathbf{Q}}[\underline{q}(t)] \cdot \underline{\hat{\omega}}(t) \quad (13)$$

as the first guess, and at each step the constraint

$$\|\underline{q}(t + \Delta t)\| = 1 \quad (14)$$

is imposed. In the test cases presented below eqs. (9)–(14) are iterated until the components of successive estimates of  $\underline{q}(t + \Delta t)$  differ by less than  $10^{-6}$ . Typically no more than three or four iterations are needed for convergence.

This integration scheme can be used for rigid bodies in a system containing "atomic" species (whose equations of motion are integrated with the standard leapfrog algorithm). These rigid bodies may even be linked to other species (including other rigid bodies) by extensible bonds. However, if a rigid body is linked to an atom or another rigid body by a bond *constraint*, the algorithm is not adequate. The bond constraint will introduce an additional force and torque on the body that can only be found *after* the integration of the unconstrained unit. The QSHAKE algorithm has been developed to deal with such cases. However, before we introduce the QSHAKE algorithm we briefly review the SHAKE algorithm for atomic systems.

## SHAKE

The first stage of the SHAKE algorithm is to propagate the unconstrained equations of motion for an atomic system. Assuming a leapfrog inte-

gration scheme,

$$\underline{v}_i\left(t + \frac{\Delta t}{2}\right) = \underline{v}_i\left(t - \frac{\Delta t}{2}\right) + (\Delta t/m_i)\underline{f}_i(t), \quad (15)$$

$$\underline{r}_i(t + \Delta t) = \underline{r}_i(t) + \Delta t \underline{v}_i\left(t + \frac{\Delta t}{2}\right), \quad (16)$$

where  $i$  labels atoms,  $\underline{v}$  is atomic velocity, and  $\underline{r}$  is atomic position. However, this does not ensure the constraints are satisfied at time  $t + \Delta t$  and a correction to atomic positions and forces is required. The constraint force (which acts along the bond vector at time  $t$ ) required to maintain the bond length at the desired value is given (to first order<sup>5</sup>) by

$$\underline{f}_{\text{const}} = \frac{\mu}{2\Delta t^2} \frac{(d^2 - d'^2)}{\underline{d}' \cdot \underline{d}} \underline{d}, \quad (17)$$

where  $\underline{d}$  is the constrained bond vector at the start of the step and  $\underline{d}'$  is the bond vector at the end of the unconstrained integration step;  $\mu$  is the reduced mass of the atoms in the constraint bond [i.e.,  $\mu^{-1} = (m_i^{-1} + m_j^{-1})$ , reflecting the fact that the bond correction depends upon the reciprocal masses of both atoms in the bond]. If the atoms concerned are not involved in any other constraint bonds, then the error in eq. (17) is of the order  $\mathcal{O}(\Delta t^2)$ , but if a network of constraints exists eq. (17) represents only a first approximation to the constraint force. In either case the corrections arising from all the holonomic constraints to the atomic positions at  $t + \Delta t$  are given by

$$\Delta \underline{r}_i(t + \Delta t) = (\Delta t^2/m_i) \sum_{\gamma} \underline{f}_{\text{const}}^{\gamma}, \quad (18)$$

where the sum over  $\gamma$  runs over all constraints involving atom  $i$ . The corrected positions are used to recalculate the  $\underline{d}'$  and eq. (17) is used again to get further corrections to the constraint forces and hence the atomic positions [eq. (18)]. This process is repeated until each constraint is within a prescribed tolerance of  $d$ . This can be imposed by iterating until all constrained bonds simultaneously satisfy

$$|d - d'|/d < 10^{-p}, \quad (19)$$

where  $p$  is the required precision. A value of  $p = 4$  is common in simulation work. Higher values have been used, but values lower than this will lead to unacceptably large energy fluctuations (and energy drift) in the course of a simulation.

## QSHAKE

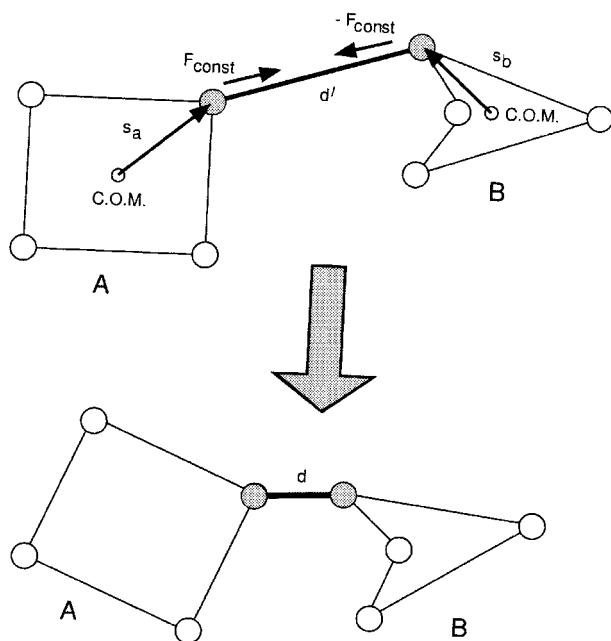
The extension of SHAKE to holonomic constraints between rigid bodies proceeds by solving the quaternion equations and the bond constraint conditions iteratively until self-consistency is obtained.

Consider Figure 1 in which two rigid bodies are linked by a constraint bond. We seek to determine the constraint force  $\underline{F}_{\text{const}}$ , which acts along the bond vector at the start of the time step, so that at the end of the integration step the two sites in the constraint bond are a distance  $d$  apart. The integration of the bodies as free units leaves the sites  $d'$  apart. Because the constraint force produces both a force and torque on the rigid units, the correction to the constrained site positions must include both the translation and rotation of each rigid body.

The *translational* contribution to the correction of the constraint site on body  $A$  is

$$\Delta \underline{r}_T = (\Delta t)^2 \underline{F}_{\text{const}} \left( \frac{1}{M_A} \right), \quad (20)$$

where  $M_A$  is the mass of rigid body  $A$ .



**FIGURE 1.** Schematic diagram of the QSHAKE algorithm. The top diagram shows two bodies after unconstrained integration. The bottom diagram shows the corrected positions after the constraint forces are taken into account. Note the constraint force acts along the bond vector at the *start* of the time step.

The torque induced by the constraint force (on body  $A$ ) is

$$\underline{\tau}_{\text{const}} = \underline{s}_a \times \underline{F}_{\text{const}}, \quad (21)$$

so the correction to the angular velocity of the body is

$$\Delta \underline{\omega}_A = \Delta t \underline{\mathbf{I}}_A^{-1} \cdot \underline{\tau}_{\text{const}}. \quad (22)$$

In implementing this it is convenient to use the transformation

$$\underline{\mathbf{I}}_A = \underline{\mathbf{R}}^\dagger \cdot \underline{\hat{\mathbf{I}}}_A \cdot \underline{\mathbf{R}} \quad (23)$$

because  $\underline{\mathbf{R}}$  has already been calculated and  $\underline{\hat{\mathbf{I}}}_A$  is diagonal. The *rotational* correction to the position of the constraint site is thus

$$\begin{aligned} \Delta \underline{r}_R &= \Delta t (\Delta \underline{\omega}_A \times \underline{s}_a), \\ &= (\Delta t)^2 F_{\text{const}} \left\{ \left( \underline{\mathbf{R}}^\dagger \cdot \underline{\hat{\mathbf{I}}}_A^{-1} \cdot \underline{\mathbf{R}} \cdot (\underline{s}_a \times \underline{\bar{e}}) \right) \times \underline{s}_a \right\}, \end{aligned} \quad (24)$$

where the bar denotes a unit vector and in particular  $\underline{\bar{e}}$  is the unit vector  $\underline{F}_{\text{const}}/F_{\text{const}} = \underline{d}/d$ .  $\Delta \underline{r}_R$  will not, in general, be parallel to  $\underline{F}_{\text{const}}$  and  $\Delta \underline{r}_T$ . If we denote the components of  $\Delta \underline{r}_R$  parallel and perpendicular to  $\underline{F}_{\text{const}}$  by the symbols  $\parallel$  and  $\perp$ , then the correction to the constrained site position on  $A$  is

$$\begin{aligned} \Delta \underline{r}_A &= (\Delta \underline{r}_T + \Delta \underline{r}_R^\parallel) + \Delta \underline{r}_R^\perp \\ &= \Delta \underline{r}_A^\parallel + \Delta \underline{r}_A^\perp. \end{aligned} \quad (25)$$

For infinitesimal rotations  $\Delta \underline{r}_A^\perp$  is zero; thus, an approximation, which is exact in the limit  $\Delta t \rightarrow 0$ , for the corrected position of the constrained site, is to take  $\Delta \underline{r}_A \approx \Delta \underline{r}_A^\parallel$ , that is,

$$\begin{aligned} \Delta \underline{r}_A &= \Delta \underline{r}_T + \Delta \underline{r}_R^\parallel = (\Delta t)^2 F_{\text{const}} \\ &\times \left\{ \frac{1}{M_A} + \left[ \left( \underline{\mathbf{R}}^\dagger \cdot \underline{\hat{\mathbf{I}}}_A^{-1} \cdot \underline{\mathbf{R}} \cdot (\underline{s}_a \times \underline{\bar{e}}) \right) \times \underline{s}_a \right] \cdot \underline{\bar{e}} \right\}. \end{aligned} \quad (26)$$

It is clear from eq. (26) that the term in braces has the dimension of reciprocal mass. This permits the definition of an effective mass,  $\mathcal{M}_A$ , for use in computing the constraint correction,

$$1/\mathcal{M}_A = 1/M_A + \left\{ \left( \underline{\mathbf{R}}^\dagger \cdot \underline{\hat{\mathbf{I}}}_A^{-1} \cdot \underline{\mathbf{R}} \cdot (\underline{s}_a \times \underline{\bar{e}}) \right) \times \underline{s}_a \right\} \cdot \underline{\bar{e}}. \quad (27)$$

Note that  $\mathcal{M}_A$  depends upon the orientation of the body with respect to the constraint bond at the start of the time step. A similar correction to (26) applies to the rigid body  $B$  attached to the other end of the constraint bond, for which a second effective mass  $\mathcal{M}_B$  may be defined as in eq. (29). (It should be noted, however, that the constraint force  $\underline{F}_{\text{const}}$  acts in the opposite direction in this case.)

With the correction eq. (26) and the defined effective mass (27), the derivation of the constraint force closely follows that of the SHAKE algorithm.<sup>5</sup> The result is the approximate formula

$$\begin{aligned} F_{\text{const}} &= \frac{\Theta}{2\Delta t^2} \frac{(d^2 - d'^2)}{\underline{d}' \cdot \underline{\bar{e}}} \cdot \underline{d} \\ &= \frac{\Theta}{2\Delta t^2} \frac{(d^2 - d'^2)}{\underline{d}' \cdot \underline{\bar{e}}}, \end{aligned} \quad (28)$$

in which the effective reduced mass  $\Theta$  for constraint linking two bodies,  $A$  and  $B$ , has been defined as follows:

$$\Theta = \mathcal{M}_A \mathcal{M}_B / (\mathcal{M}_A + \mathcal{M}_B). \quad (29)$$

With this reduced mass and the known bond vectors  $\underline{d}$  and  $\underline{d}'$ , the constraint force can be estimated.

The full QSHAKE procedure is therefore

1. evaluate the unconstrained forces and torques from the atomic positions at time  $t$  (at this stage  $F_{\text{const}} = 0$ );
2. integrate the equations of motion to  $t + \Delta t$  [eqs. (3)–(14)];
3. check the convergence of constrained bonds to within the precision required; and
4. if all the bond constraints are not satisfied, then calculate increments to the constraint forces from the atomic positions at  $t + \Delta t$  [using eq. (28)] and go back to step 2 with the forces in eq. (3) now including the current estimate of the constraint forces.

Note that at each iteration a “correction” to the constraint force is calculated (with the initial approximation  $F_{\text{const}} = 0$ ) and that each iteration requires the rigid body equations of motion to be reintegrated from  $t$  to  $t + \Delta t$  before the next iteration may be made.

Although QSHAKE has a strong resemblance to SHAKE there are some important differences. First, because of the approximations made in eq. (26), eq. (28) is no longer correct to order  $\mathcal{O}(\Delta t^2)$ . However,

for the simplest case (two rigid bodies linked by one constraint bond) only two (at most three) iterations are normally necessary for convergence. Second, the reduced mass term needed by QSHAKE [eqs. (25), (26)] needs to be reevaluated every time step (although not every iteration). Third, with SHAKE it is most efficient to cycle through all constraints, adjusting atomic positions before the next constraint is considered. With QSHAKE the corresponding strategy would require reintegrating the quaternion equations of motion for the two constrained rigid bodies before considering the next constraint. Because the integration step is considerably more expensive than in the atomic case, this strategy is less efficient than evaluating all the constraint forces independently at each cycle and reintegrating once with the combined forces and torques of all the constraints acting on the body.

Finally, we note that QSHAKE reduces to the SHAKE algorithm if the rigid bodies are replaced by point atoms. (In such a case, care must be taken to avoid the singularity arising from  $\mathbf{I}^{-1}$ , even though the  $s_\alpha$  are zero.) Hence, as with SHAKE, convergence will be slower if extended networks of linked bodies are involved. However, as will be seen in the test case below, QSHAKE can be approximately an order of magnitude faster than SHAKE in such situations.

## THERMOSTATS AND BAROSTATS

It is straightforward to couple the linked rigid body equations of motion to a thermostat and/or barostat. The thermostat is coupled to both the translational and rotational degrees of freedom, and so both the translational and rotational velocities are propagated in an analogous manner to the thermostatted atomic velocities. The barostat, however, is coupled only to the translational degrees of freedom.

## Test Cases

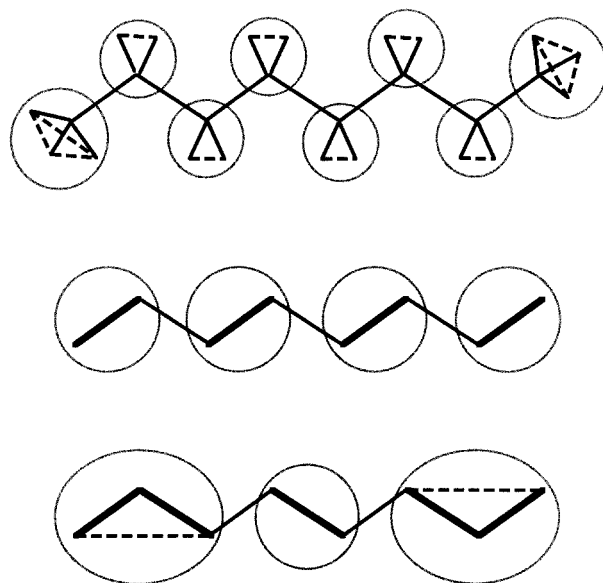
QSHAKE has been implemented in the macromolecular simulation package DL-POLY.<sup>10</sup> Three test cases on the linear alkane octane are reported. This molecule is one for which SHAKE is known to be comparatively expensive. In the test cases the atomic sites are unchanged and the intermolecular potential is described solely in terms of Lennard-Jones pair interactions. The intramolecu-

lar bond angle and torsion terms are taken from the AMBER force field.<sup>11</sup> Each test case has a different constraint network. The three models are (Fig. 2): the all atom model, the united atom model, and united atom with two bond angle constraints.

Each test system contained 64 molecules and in each case the molecular dynamics cell was a cube with a length of 27.2 Å. Lennard-Jones interactions were truncated at 10 Å and periodic boundary conditions applied. Results are for equilibrated simulations at 400 K and 0 bar with averages taken over 1 ps of simulation.

### ALL ATOM MODEL

In this model all 26 atoms (each with 3 degrees of freedom) of  $\text{C}_8\text{H}_{18}$  are treated explicitly. All C—H bonds are constrained to 1.09 Å and all H—C—H angles to 109.5°. When SHAKE is used the structure of the terminal  $\text{CH}_3$  groups is maintained by six constraint bonds (three C—H and three H—H constraints) and the  $\text{CH}_2$  units by three constraint bonds (two C—H and one H—H constraint). In addition, all seven C—C bonds are



**FIGURE 2.** The models of octane. In each case solid lines represent bonds between atoms, dotted lines additional constraints between 1–3 neighbors to fix bond angles. The ellipsoids enclose sites treated as part of the same rigid body by QSHAKE. Top: The all atom model,  $\text{C}_8\text{H}_{18}$ ; C—C bonds and C—H bonds shown. Middle: The united atom model (model 1),  $\text{CH}_3$  and  $\text{CH}_2$  groups taken as single entities; C—C bonds shown. Bottom: The united atom model (model 2) in which the terminal bond angles are constrained.

constrained to  $1.526 \text{ \AA}$ , giving a total of 37 constraints. The molecule has  $26 \times 3 - 37 = 41$  degrees of freedom.

In the QSHAKE description each  $\text{CH}_3$  and  $\text{CH}_2$  unit is described as a rigid body (with 6 degrees of freedom) and these units are linked by constraints along the C—C bonds. This description thus has eight rigid bodies and seven constraints, which also yields  $41 (= 8 \times 6 - 7)$  degrees of freedom.

Simulations using a time step of 2, 4, and 6 fs are reported. Only the QSHAKE algorithm was stable with the 6-fs time step; neither algorithm was stable with an 8-fs time step.

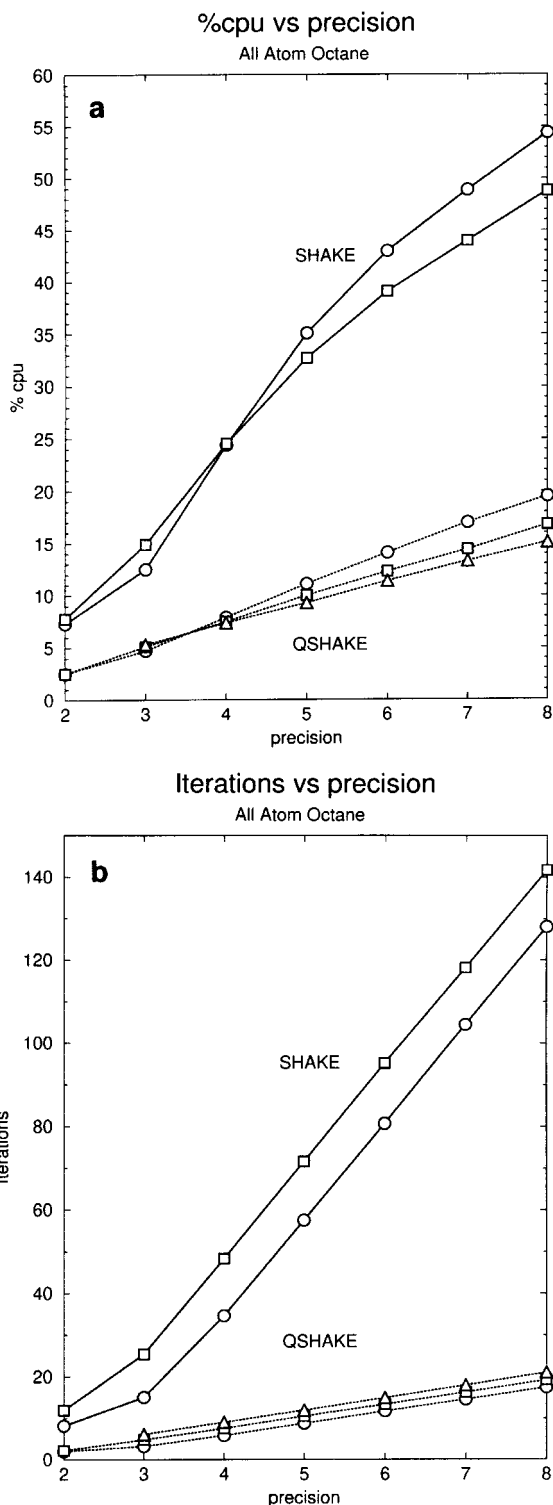
Figure 3 shows two performance plots for each algorithm. The most crucial is (a), the relative amount of CPU time that the integration algorithm requires. The lower this is, the more efficient the algorithm. This is shown as a function of the precision over the range  $p = 2$ –8. Figure 3a clearly shows that QSHAKE is substantially more efficient than SHAKE over the entire range studied. At  $p = 4$  QSHAKE requires only 8% of the total CPU time while SHAKE requires 25%. (At this precision QSHAKE is 3.8 times faster than SHAKE; at  $p = 6$  QSHAKE is 11 times faster.) QSHAKE also has the advantage that it permits larger time steps than are possible with SHAKE.

It should be noted that for a given value of  $p$ , QSHAKE will also give lower energy fluctuations than SHAKE because all interatomic distances in the rigid bodies are exactly (to machine precision) constant. In the SHAKE algorithm these are only constant to within the precision  $p$ .

Although each iteration of QSHAKE is more costly than SHAKE (because the quaternion equations of motion must be solved repeatedly), QSHAKE is more efficient. Although there are some gains made because the number of constraint bonds is considerably reduced in the QSHAKE description (7 as opposed to 37), the principal reason for the efficiency is because QSHAKE circumvents the occurrence of the triangles of constraints that are so poorly convergent for SHAKE. The effect of such triangles on the relative performance of the two algorithms is demonstrated by the next two test cases in which octane is treated at the united atom level.

### UNITED ATOM OCTANE: MODEL 1

In this model each  $\text{CH}_3$  and  $\text{CH}_2$  is treated as a single spherical site with a mass of 15.034 or 14.026 amu, respectively. In the SHAKE algorithm each



**FIGURE 3.** All atom octane. Data points pertain to 1-ps simulations with a time step of (○) 2 fs, (□) 4 fs, and (△) 6 fs. (a) The percent of the total CPU time per time step taken by the integration schemes (propagation of equations of motion plus application of constraints); (b) the corresponding number of iterations required by each constraint method.

unit is connected by a C—C constraint bond of 1.526 Å. For the QSHAKE algorithm the molecule is considered as four (linear) rigid bodies connected by three constraint bonds (Fig. 2). In both descriptions the molecule has  $8 \times 3 - 7 = 4 \times 5 - 3 = 17$  degrees of freedom. Simulations are reported with time steps of 5, 10, and 20 fs.

In this example, although QSHAKE requires fewer iterations than SHAKE (Fig. 4b), QSHAKE is the less efficient of the two algorithms (Fig. 4a). This is simply because the treatment of a two atom rigid body by quaternions is considerably more expensive than SHAKE-ing a bond.

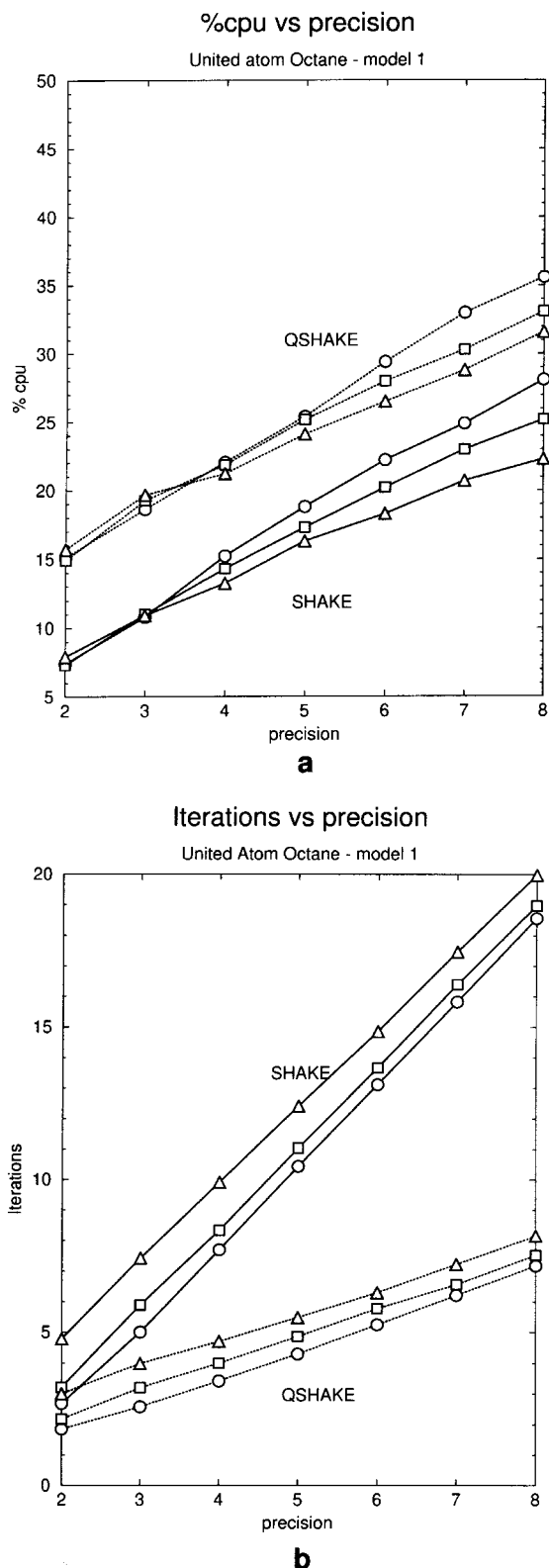
### UNITED ATOM OCTANE: MODEL 2

In this system united atom octane (model 1) is modified so that the terminal angles are constrained to 109.5°. For the SHAKE algorithm this means adding two additional bond constraints (between atoms 1 and 3, and between atoms 6 and 8). For QSHAKE this means describing the molecule as three rigid bodies linked by two constraint bonds, the constraints being between atoms 3 and 4 and between 5 and 6.

In this example triangles of constraints reappear, and for all normal working values of  $p$  QSHAKE is more efficient than SHAKE. QSHAKE requires fewer iterations for convergence and less CPU per time step (Fig. 5). Comparison of Figures 4 and 5 shows that the additional constraints in the second united atom model has little effect on the performance of QSHAKE, but they increase the number of SHAKE iterations required by about 50%.

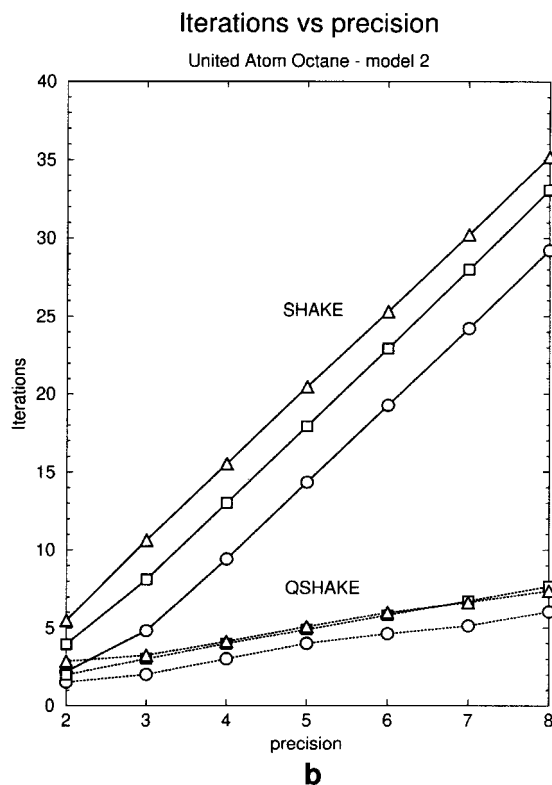
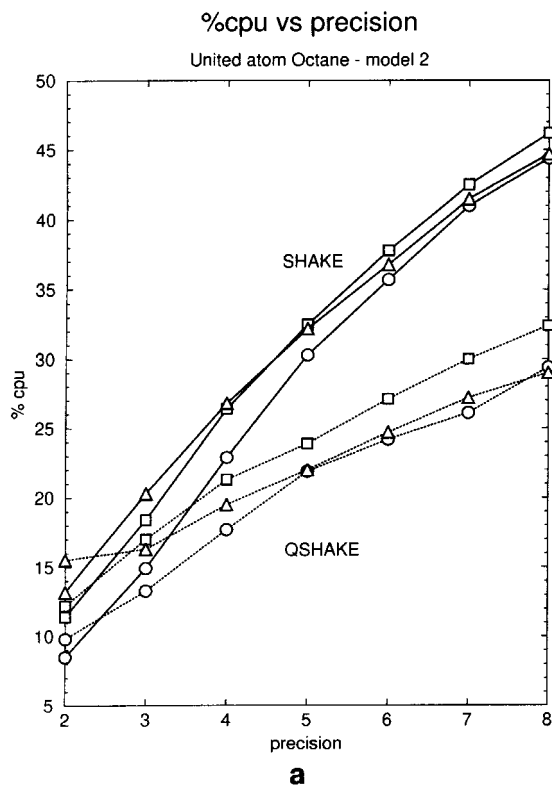
### FURTHER COMMENTS

Figures 3–5 have some features in common for all the test systems. First, the greater the time step, or the higher the precision, the greater the number iterations needed to guarantee convergence. This is expected for both SHAKE and QSHAKE, but it is noticeable that QSHAKE scales much better than SHAKE with an increasing time step and increasing precision. Second, it will be noticed that as the time step increases, the relative amount of CPU consumed by the constraint algorithm decreases, even though more iterations are required. This is simply a reflection of the fact that in our code the updating of neighborhood lists is determined by the atomic displacements (not the number of time steps) since the previous update. Thus, simulations with larger time steps will update neighbor-



**FIGURE 4.** United atom octane model 1. Annotation as per Figure 3 except that results are for time steps of (○) 5 fs, (□) 10 fs, and (△) 20 fs (triangles).





**FIGURE 5.** United atom octane model 2. Annotation as per Figure 4.

hood lists more frequently (in terms of number of time steps between updates) than simulations with smaller time steps. Consequently, the mean workload per step (ignoring the demands of the integration routines) increases with increasing time steps.

### Discussion: Comments on Applicability of QSHAKE

It is clear from the derivation of QSHAKE that the method represents an extension of the standard SHAKE algorithm. The method is therefore applicable to any system in which bond constraints are practical, but it additionally offers the possibility of greatly reducing the complexity of the constraint prescription and improving the overall efficiency whenever parts of a molecule may be reasonably represented by rigid collections of atoms. Its main advantage over SHAKE is in systems where connected triangles of constraints are required. Such triangles occur wherever rigid planar or 3-D substructures are present. Examples of such systems are many; aromatic and other ring structures appear in synthetic polymers (e.g., styrene), amino acids (thyroxine, tyrosine, tryptophan, etc.), and nucleic acids (adenine, cytosine, guanine, thymine) and, depending on the nature of the simulation required, may be specified as rigid entities treatable by QSHAKE. QSHAKE should thus find wide use in atomistic simulations of polymeric materials such as peptides, nucleotide fragments (DNA and RNA), synthetic polymers, and where all atom descriptions of alkyl chains are used.

As mentioned in the Introduction, QSHAKE does not allow rigid units to share single atoms (pivots) or bonds (hinges) with other rigid units. It is possible to treat such systems using the matrix inversion method of Kneller and Hinsien.<sup>8</sup> Pivots and hinges are particularly useful in modeling chain molecules and fused ring structures; however, it does not follow that the dynamically restricted model that results is suitable for all studies. (Of course QSHAKE will handle a pivot system in which one of the rigid bodies is a rigid diatomic. There also appears to be no reason why two rigid bodies linked together by more than one constraint bond cannot be handled by QSHAKE, but this needs to be explored, however.) QSHAKE's main advantages are that it is particularly simple to implement and has no potentially large and therefore computationally expensive matrices to

invert at each time step as in the Kneller and Hinsen method.

Finally, although we developed QSHAKE in an atomistic frame, the generalization to include constraint bonds between nonspherical particles (e.g., Gay–Berne ellipsoids) is straightforward.

---

## Conclusions

The QSHAKE algorithm has been introduced and shown to be a significant improvement on SHAKE for applying holonomic constraints in the simulation of semiflexible molecules. The method has potentially wide applicability in molecular simulation and is easily incorporated into existing programs employing the SHAKE algorithm for constraint bonds.

---

## References

1. A. Rahman and F. H. Stillinger, *J. Chem. Phys.*, **55**, 3336 (1971).
2. J. Barojas, D. Levesque, and B. Quentrec, *Phys. Rev.*, **A7**, 1092 (1973).
3. D. J. Evans, *Mol. Phys.*, **34**, 1745 (1977).
4. D. Fincham, *Mol. Simul.*, **8**, 165 (1992).
5. J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, *J. Comput. Phys.*, **23**, 327 (1977).
6. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, Clarendon Press, Oxford, U.K., 1989.
7. H. C. Andersen, *J. Comput. Phys.*, **54**, 24 (1983).
8. G. Kneller and K. Hinsen, *Phys. Rev. E.*, **50**, 1559 (1994).
9. H. Goldstein, *Classical Mechanics*, 2nd ed., Addison–Wesley, Reading, MA, 1980.
10. W. Smith and T. R. Forester, *J. Mol. Graph.*, **14**, 136 (1996).
11. S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case, *J. Comput. Chem.*, **7**, 230 (1986).